Personalized Automatic Sleep Staging with Single-Night Data: a Pilot Study with KL-Divergence Regularization

Huy Phan[1], Kaare Mikkelsen[2], Oliver Y. Chén[3], Philipp Koch[4], Alfred Mertins[4], Preben Kidmose[2], Maarten De Vos[3,5]

[1]School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[2]Department of Engineering, Aarhus University, Denmark
[3]Institute of Biomedical Engineering, University of Oxford, UK
[4]Institute for Signal Processing, University of Lübeck, Germany
[5]Department of Electrical Engineering, KU Leuven, Belgium

E-mail: h.phan@qmul.ac.uk

April 2020

Abstract.

Objective: Brain waves vary between people. This work aims to improve automatic sleep staging for longitudinal sleep monitoring via personalization of algorithms based on individual characteristics extracted from the first night of data. *Approach:* As a single night is a very small amount of data to train a sleep staging model, we propose a Kullback-Leibler (KL) divergence regularized transfer learning approach to address this problem. We employ the pretrained SeqSleepNet (i.e. the subject independent model) as a starting point and finetune it with the single-night personalization data to derive the personalized model. This is done by adding the KL divergence between the output of the subject independent model and the output of the personalized model to the loss function during finetuning. In effect, KL-divergence regularization prevents the personalized model from overfitting to the single-night data and straying too far away from the subject independent model. *Main results:* Experimental results on the Sleep-EDF Expanded database with 75 subjects show that sleep staging personalization with a single-night data is possible with help of the proposed KL-divergence regularization. On average, we achieve a personalized sleep staging accuracy of 79.6%, a Cohen's kappa of 0.706, a macro F1-score of 73.0%, a sensitivity of 71.8%, and a specificity of 94.2%. *Significance:* We find both that the approach is robust against overfitting and that it improves the accuracy by 4.5 percentage points compared to non-personalization and 2.2 percentage points compared to personalization without regularization.

1. Introduction

The increased awareness of the important role of sleep in protecting our mental and physical health [1] has been translated in an increased demand in personal sleep monitoring tools. For such purpose, automating sleep scoring is vital and indispensable

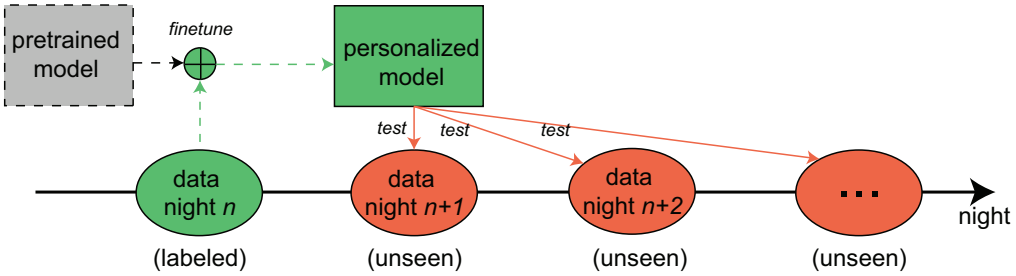

Figure 1: Personalization with single-night data: a pretrained model is finetuned with the labeled data of night n of an undividual and yields the personalized model which is tested on the same individual's unseen data of nights $n+1$, $n+2$, ...

since manual scoring is simply too expensive, time-consuming, and labor-intensive [2, 3]. The advance of machine learning, deep learning in particular, coupled with the availability of large sleep databases [4, 5, 6] has stimulated a new wave of interest in developing automatic sleep staging methods. In fact, machine performance in sleep staging has progressed significantly, being on par with manual scoring by sleep experts, thanks to recent methods based on deep learning [7, 8, 6].

The above-mentioned state-of-the-art classification performance is only possible using supervised learning. That is, we need data to be recorded and manually labeled from a cohort of subjects, followed by model training based on the labeled data. In fact, the recent expert-level performance is only obtainable with a large cohort (i.e. hundreds or thousands of subjects) [7, 6]. Collecting and manually scoring a large amount of sleep data is a vast burden, particularly for wearable EEG devices like in-ear EEG [9] or around-the-ear EEG [10, 11], in which case the work load is increased by the need for an added PSG for reference. Utilizing and including available sleep data for training a sleep staging algorithm in novel settings is not easy, due to channel mismatch caused by differences in channel layouts, electrode placements, recording devices and software, preprocessing procedure, normalization parameters, clinical cohort characteristics, etc. [7]. The work in [12, 7] proposed a transfer learning approach to circumvent the above-mentioned channel mismatch and enable knowledge transfer from a large dataset to a small cohort, making a deep learning model for a different, specific setting with low amount of data possible. However, such a transfer learning approach still requires data from a dozen of subjects to succeed. Although collecting and labeling this relatively small amount of sleep data would not be a big problem, here we want to push this data constraint to its extreme and question whether it is possible to adapt a pretrained model with single-night data of a particular subject, i.e. personalization, even without knowing in which setting the data is recorded. By personalization, we mean the parameters of the pretrained model are adapted to an individual's data to convert into a personalized model which is later tested on the same individual's future unseen data as illustrated in Figure 1. If personalization with single-night data in an unknwon setting is possible, it would be convenient for one to build a model for personalized sleep monitoring using

his/her very own minimal data recorded with a particular device. It is equally important and necessary when privacy and security become a serious concern [13, 14, 15], and thus, owning EEG data from others to form a cohort for transfer learning [7] would be more and more difficult. An additional and very important benefit of personalization is that it has previously been shown that automatic sleep scoring becomes more accurate when the classifier can focus on the peculiarities of the individual (see [16] and [9]). This is especially the case when using non-standard EEG montages, for instance in in-ear EEG and around-the-ear EEG. It should be noted that this personalization problem is different from that in [17] in which a cohort of subjects is known and a model is trained on the cohort before personalizing for a subject in the same cohort. Here, we assume there is no information about the cohort or recording settings but only a single-night data of a target subject is available.

Building a deep-learning model using single-night data is challenging. First, the model can easily overfit the data regardless of whether we train a model from scratch or finetune a pretrained model [7]. Second, different subjects are expected to have varying convergence/overfitting rate when training/finetuning the personalized model. Therefore, we do not know when the model will start overfitting, as we do not have validation data at hand for model selection as in the case of a cohort [7, 12]. Third, regular data normalization cannot be done as a cohort's statistics are unknown. In this work, we take on this 'personalization with single-night data' challenge and propose an approach based on transfer learning to deal with it. We employ the pretrained SeqSleepNet [7, 18] (i.e. the subject independent (SI) model), and finetune it with single-night data from a single subject from an unknown cohort to accomplish personalization. Note that, the source-domain cohort which was used for pretraining the model is also assumedly unknown. To remedy the overfitting problem, KL-divergence between the output of the SI model and the personalized model is introduced to regularize the network. The KL-divergence regularization, in effect, prevents the personalized model from drifting too far away from the SI model. Once we get rid of overfitting, model selection is no longer an issue as we can keep finetuning the SI model as long as we need. Experiments on 75 subjects of the Sleep-EDF Expanded database [19, 20] show that KL-divergence regularized personalization with single-night data is robust against overfitting and achieves an average sleep staging accuracy of 79.6%, improving 4.5 and 2.2 percentage points over non-personalization and personalization without KL-divergence regularization, respectively.

2. Material

We used the Sleep-EDF Expanded database (Sleep Cassette subset, version 2018) [19, 20] in this study. This database consists of 78 healthy Caucasian subjects aged 25-101. It is particularly suitable for this study as there are 75 out of 78 subjects with two subsequent day-night PSG recordings collected for each. Three subjects (subjects 13, 36, and 52) whose one recording was lost due to device failure were excluded from the personalization

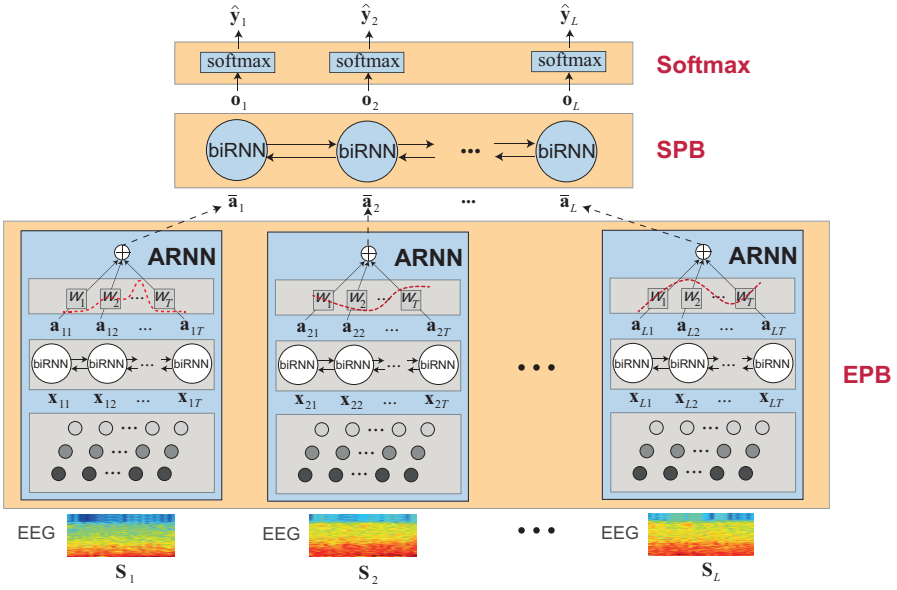

Figure 2: Illustration of SeqSleepNet which is composed of three components: epoch processing block (EPB), sequence processing block (SPB), and Softmax. Image adapted from [18].

experiments. Manual scoring was done by sleep experts according to R&K standard [3] and each 30-second PSG epoch was labeled as one of eight categories W, N1, N2, N3, N4, REM, MOVEMENT, UNKNOWN. We merged N3 and N4 into a single stage N3 and excluded MOVEMENT and UNKNOWN categories as in previous experiments in earlier versions of the database [21, 22, 23, 8, 24]. We used the Fpz-Cz EEG channel sampled at 100 Hz in this study. As different portions of this database have been used in the literature, it should be stressed that we only made use of the *in-bed* parts (from *lights off* time to *lights on* time) recommended in [25, 21] and adopted in many existing works [22, 23, 24, 7, 26, 27, 17, 28].

3. Methods

3.1. Sequence-to-Sequence Sleep Staging with SeqSleepNet

SeqSleepNet, recently proposed in [18], has demonstrated state-of-the-art performance on several sleep databases [18, 12] and its suitability for transfer learning tasks [12, 7]. We employ it in this work to study sleep-staging personalization. As a sequence-to-sequence sleep-staging model [18], SeqSleepNet learns to maximize the conditional probability $p(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L \,|\, \mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_L)$ [18]. In other words, it receives a sequence of L consecutive epochs $(\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_L)$ and classifies them at once into a sequence of corresponding sleep stages $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L)$, where $\mathbf{y}$ is a one-hot encoding vector.

To be fed into the network, the EEG signal of a 30-second epoch is transformed into a time-frequency image $\mathbf{S} \in \mathbb{R}^{F \times T}$ obtained via short-time Fourier transform (STFT), where F is the number of frequency bins and T is the number of time instances (cf.

Section 4). The network is composed of three main components: epoch processing block (EPB), sequence processing block (SPB), and Softmax, as illustrated in Figure 2.

EPB. EPB is essentially an attention-based RNN (ARNN) [27] that is shared by all epochs in the input sequence for short-term (i.e. intra-epoch) sequential modelling. The ARNN subnetwork consists of a *filterbank* layer [26], a bidirectional RNN realized by long short-term memory (LSTM) cells [29] with recurrent batch normalization [30], and a self-attention layer [31]. The trainable filterbank layer with M filters is designed to smooth and reduce the frequency dimension of each epoch $\mathbf{S}$ from F to M, where $M < F$ [26]. The resulting image is then treated as a sequence of T local feature vectors $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ (corresponding to T spectral columns) which is encoded by the bidirectional RNN into a sequence of output vectors $(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_T)$. The self-attention layer [31] is trained to produce attention weights $(w_1, w_2, \ldots, w_T)$ and combines the output vectors into a single feature vector $\bar{\mathbf{a}} = \sum_{t=1}^{T} w_t \mathbf{a}_t$ to represent the epoch $\mathbf{S}$.

SPB. SPB is a bidirectional RNN for long-term (i.e. inter-epoch) sequential modelling. Similar to the RNN in EPB, this RNN is also realized by LSTM cells [29] with recurrent batch normalization [30]. After EPB, the input sequence $(\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_L)$ has been converted into a sequence of feature vectors $(\bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2, \ldots, \bar{\mathbf{a}}_T)$. In turn, the bidirectional RNN iterates over the sequence of induced feature vectors and encode it into the sequence of output vectors $(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_L)$.

Softmax. Given the sequence of output vectors $(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_L)$, classification eventually takes place at the Softmax component to produce the sequence posterior probabilities $(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_L)$, where $\hat{\mathbf{y}}_l$ coresponds to the epoch at index l, $1 \leq l \leq L$, in the input sequence. Similar to the SeqSleepNet+ variant in [7], the softmax layer is shared between all epochs.

The network is trained end-to-end to minimize the sequence classification loss over all N training sequences in the training data:

$$\begin{aligned} E(\Theta) &= -\frac{1}{L}\sum_{n=1}^{N}\sum_{l=1}^{L}\mathbf{y}_{nl}\log\left(\hat{\mathbf{y}}_{nl}\left(\Theta\right)\right) + \frac{\lambda}{2}\|\Theta\|_2^2 \\ &= -\frac{1}{L}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{c\,\in\,\mathcal{C}}\mathbb{I}(y_{nl}=c)\log P_{\Theta}(\hat{y}_{nl}=c) + \frac{\lambda}{2}\|\Theta\|_2^2, \end{aligned} \tag{1}$$

where $\mathcal{C} = \{\mathrm{W}, \mathrm{N1}, \mathrm{N2}, \mathrm{N3}, \mathrm{REM}\}$ is the set of all possible sleep stages. In (1), $\mathbb{I}(\cdot)$ is the indicator function, y_{nl} and $\hat{y}_{nl}$ denotes the ground-truth and output discrete labels of the l^{th} epoch in the n^{th} sequence, respectively. Θ denotes the trainable parameters of the network and λ is the coefficient of the ℓ_2-norm regularization term.

3.2. KL-Divergence Regularization for Personalization

Given the small amount of data (one night) it is not feasible to train a deep learning model like SeqSleepNet from scratch. As mentioned before, we, therefore, pursue a transfer learning approach similar to [7, 12] for personalization. We use the pretrained

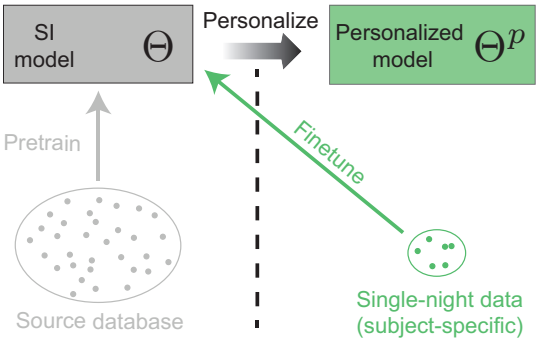

Figure 3: Illustration of sleep personalization with single-night data. The subject independent (SI) model Θ, which is pretrained with a source-domain database (assumedly unknown), is finetuned on the single-night data of a target subject to derive the personalized model Θ^p.

SeqSleepNet model from [7], which was pretrained using the C4-A1 EEG data from 200 subjects (686,610 epochs in total) of the Montreal Archive of Sleep Studies (MASS) database [5] (i.e. the source database), as the subject independent (SI) model denoted by Θ. We would like to remind the reader that the MASS cohort is assumedly unknown here. The SI model Θ then serves as the starting point and is finetuned using the single-night data of a target subject to derive the personalized model, denoted by Θ^p, as illustrated in Figure 3. Note that channel mismatch is expected between the source-domain MASS database and the target subject's personalization data, and finetuning is supposed to address both channel mismatch and personalization. We investigate four finetuning strategies $\{$*All*, *EPB+Softmax*, *SPB+Softmax*, *Softmax*$\}$ similar to those in [7, 12]. When components of the pretrained network (i.e. the entire network, EPB+Softmax, SPB+Softmax, or Softmax depending on the finetuning strategies) are finetuned, their weights are adapted with the personalization data while the rest remains fixed.

The study in [7] showed that sleep transfer learning requires roughly at least ten subjects' data, leaving personalization with the single-night data of a target subject exposed to the substantial risk of overfitting. In fact, we experimentally see that the personalized model tends to overfit the personalization data very easily. Moreover, there exists no viable way to select the right model during finetuning before overfitting starts. One may leave out a portion of the one-night data for validation. However, since the validation data is distributed very similarly to the finetuning data, this leave-out validation data is also overfitted easily and cannot be used to identify overfitting. To remedy overfitting, we propose to regularize the sequential classification loss function in (1) with the KL divergence between the posterior probability outputs of the SI model Θ and the ones from the personalized model Θ^p, which constrains the personalized model not to stray too far away from the SI model [32]. Given an input sequence $(\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_L)$, KL divergence between the outputs of the two models reads:

$$D_{\mathrm{KL}} = \frac{1}{L} \sum_{l=1}^{L} \sum_{c \in \mathcal{C}} P_{\Theta}(\hat{y}_l = c) \log \left(\frac{P_{\Theta}(\hat{y}_l = c)}{P_{\Theta^p}(\hat{y}_l = c)} \right). \tag{2}$$

The KL-divergence regularization is added into the sequential classification loss

function in (1) to form the loss function for personalization:

$$E(\Theta^p) = -(1-\alpha)\frac{1}{L}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{c\in\mathcal{C}}\mathbb{I}(y_{nl}=c)\log P_{\Theta^p}(\hat{y}_{nl}=c) + \frac{\lambda}{2}\|\Theta^p\|_2^2 + \alpha\frac{1}{L}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{c\in\mathcal{C}}P_{\Theta}(\hat{y}_{nl}=c)\log\left(\frac{P_{\Theta}(\hat{y}_{nl}=c)}{P_{\Theta^p}(\hat{y}_{nl}=c)}\right), \quad (3)$$

where $\alpha \in [0, 1]$ is the KL-divergence regularization coefficient, regulating how far the personalized model Θ^p deviates from the SI model Θ. When $\alpha = 0$, the KL-divergence regularization is cancelled out and the personalization turns out to be the same as regular finetuning in [7, 12]. In this case, the pretrained SI model is adapted solely on the personalization data. In contrast, when $\alpha = 1$, we trust the pretrained SI model completely and ignore all the new information of the personalization data. Since the term $\alpha\frac{1}{L}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{c\in\mathcal{C}}P_{\Theta}(\hat{y}_{nl}=c)\log P_{\Theta}(\hat{y}_{nl}=c)$ in the KL-divergence regularization term in (3) does not depend on the personalized network Θ^p, the KL-divergence regularized loss function can be simplified as:

$$E'(\Theta^p) = -(1-\alpha)\frac{1}{L}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{c\in\mathcal{C}}\mathbb{I}(y_{nl}=c)\log P_{\Theta^p}(\hat{y}_{nl}=c) + \frac{\lambda}{2}\|\Theta^p\|_2^2 - \alpha\frac{1}{L}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{c\in\mathcal{C}}P_{\Theta}(\hat{y}_{nl}=c)\log P_{\Theta^p}(\hat{y}_{nl}=c). \quad (4)$$

It turns out that the loss function for personalization in (4) consists of two terms: (1) the cross-entropy between the output of the personalized model Θ^p and the ground-truth, and (2) the cross-entropy between the output of the personalized model Θ^p and the output of the pretrained SI model Θ. As a result, model personalization is equivalent to changing the target distribution from the unknown source-domain database (the MASS database used for pretraining) to a linear interpolation of the source-domain data distribution and the personalized data distribution [32]. This interpolation prevents the network from overfitting the personalization data.

4. Experimental Setup

For each of the 75 subjects with two day-night recordings of the Sleep-EDF Expanded database, we conducted finetuning of the pretrained SeqSleepNet [7] using the data from the first night and evaluating the personalized model on the data from the second night. We experimented with different values for the KL-divergence regularization coefficient α in the set {0, 0.2, 0.4, 0.6, 0.8} to investigate its influence. Note that, when $\alpha = 0$, we excluded the KL-divergence regularization completely. This case is considered the baseline for comparison with the proposed approach.

The EEG signal was divided into 30-second epochs. Each epoch was transformed into a log-magnitude time-frequency image by the following procedure: the signal

Table 1: Performance on regular (scratch) training setup via 10-fold cross validation.

System	Data portion	Overall metrics				
		Acc.	κ	MF1	Sens.	Spec.
SeqSleepNet	*in-bed* only	79.1	0.708	74.6	74.2	94.2
DeepSleepNet [8]	*in-bed* only	78.5	0.702	75.3	75.0	94.1
SeqSleepNet	*in-bed* ± 30 min	82.6	0.760	76.4	76.3	95.4
SleepEEGNet [33]	*in-bed* ± 30 min	80.0	0.730	73.6	–	–

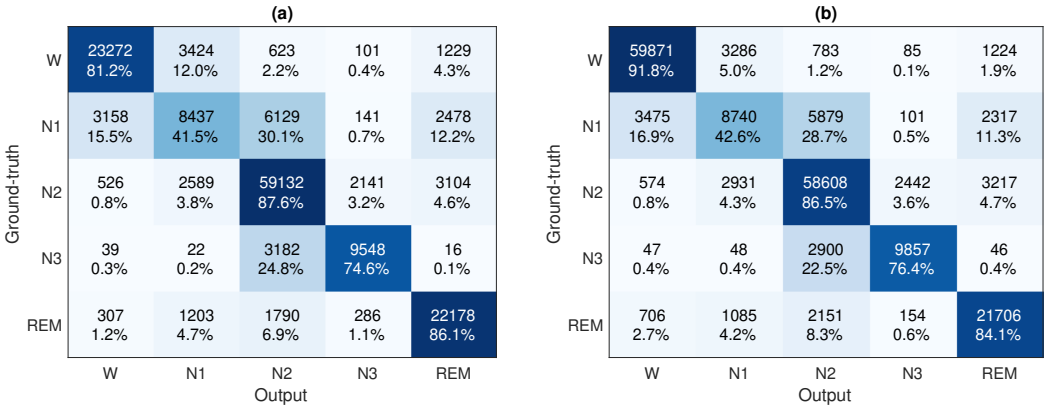

Figure 4: Confusion matrices obtained by SeqSleepNet. (a) *in-bed* data only, (b) *in-bed* data ± 30 min.

was divided into two seconds windows with 50% overlap, multiplied with a Hamming window, transformed to the frequency domain by means of a 256-point Fast Fourier Transform (FFT), and the amplitude spectrum was log-transformed. This resulted in an image of size $F \times T$ where $F = 129$ (the number of frequency bins) and $T = 29$ (the number of spectral columns).

5. Results

5.1. SeqSleepNet's performance on regular training setting.

SeqSleepNet [18, 7] requires the data to be normalized to zero mean and unit standard deviation [18, 7]. Unfortunately, in our case neither the source-domain cohort (i.e. the MASS cohort) nor the target subject's cohort (i.e. the Sleep-EDF cohort) are known. We, therefore, cannot normalize the personalization data using the cohort's statistics. In addition, we experimentally found that model personalization is sensitive to differences in magnitude of data between two nights, and per-subject data normalization resulted in poor performance in some subjects with such substaintial magnitude difference. To rule out this difference, we alternatively performed *per-night normalization* in which data of one night recording was normalized by its mean and standard deviation.

The implementation was based on the Tensorflow framework [34]. The pretrained

SeqSleepNet was parametrized similarly to the one in [7] and used a sequence length of $L = 20$. For personalization, the pretrained SeqSleepNet was finetuned on the single-night finetuning data for 50 finetuning epochs and the performance was recorded every 5 finetuning epochs. Finetuning was performed using the Adam optimizer [35] with a learning rate of 10^{-4}.

SeqSleepNet has been reported to achieve state-of-the-art performance on the MASS database [5] (i.e. the source domain used for pretraining) and the earlier version of the Sleep-EDF Expanded database with 20 subjects [19, 20]. It is worth assessing its performance on the experimental database on a regular (scratch) training setup. To this end, we conducted 10-fold cross validation on all 78 subjects. At each iteration, 7 subjects were left out for validation (i.e. model selection). During training, the network achieving the best overall accuracy on the validation subjects was retained for evaluation on the test subjects. The results of 10 cross-validation folds were pooled to calculate the overall metrics, including accuracy, macro F1-score (MF1) [36], Cohen's kappa (κ) [37], sensitivity, and specificity. Beside SeqSleepNet we also implemented the end-to-end variant of the popular DeepSleepNet [8, 18] for comparison. In addition, we include results for another common usage of the database in which 30 minutes of data before and after in-bed parts are additionally included. The performance is shown in Table 1 in which SeqSleepNet not only obtains better performance than the DeepSleepNet counterpart but also outperforms the most recent results in [33] on this latest version of the Sleep-EDF Expanded database. The accuracy of the sleep stages is also shown in the confusion matrices in Figure 4.

5.2. Influence of KL-divergence regularization.

It should be emphasized again that, different from the regular-setting experiment in Secion 5.1, only 75 subjects with two recordings were used for the personalization experiment and three subjects with one recording were excluded. The effect of KL-divergence regularization in avoiding overfitting for model personalization is exhibited in Figure 5(a) when α takes different values in $\{0, 0.2, 0.4, 0.6, 0.8\}$. Without KL-divergence regularization (i.e. $\alpha = 0$), the average accuracy of the personalized models on 75 target subjects starts declining after 5 finetuning epochs when the models most likely start overfitting the personalization data. The overfitting appears to get worse and worse with ongoing finetuning process as the average accuracy keeps decreasing. When being regularized with KL-divergence (i.e. $\alpha > 0$), the pattern of the average accuracy curve is gradually reversed when α increases, exhibiting a negligible downward tendency when $\alpha = 0.2$, plateauing after 25 finetuning epochs with $\alpha = 0.4$, and trending upward with larger values for α.

The results in Figure 5(a) also indicate that α plays the role of a trade-off parameter between the pretrained SI model and the purely personal one. When α is set small, we allow the personalized model to aggressively fit to the personalization data at the risk of severe overfitting. In contrast, when α is large, the personalized model is conservatively

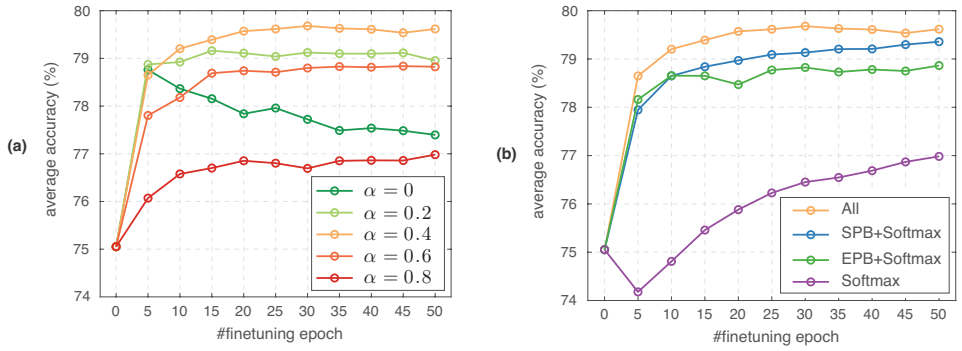

Figure 5: (a) Variation of average accuracy of 75 target subjects during finetuning (*All* strategy) with different values of α . (b) Variation of average accuracy of 75 target subjects during finetuning with different finetuning strategies (α was fixed to 0.4).

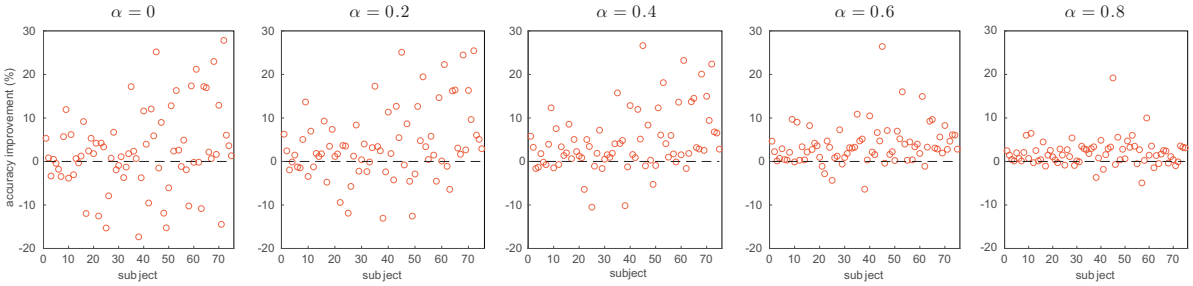

Figure 6: Individual accuracy improvements of 75 target subjects after 50 finetuning epochs when α takes different values in {0, 0.2, 0.4, 0.6, 0.8} (*All* finetuning strategy was employed).

tied to the SI model and has less freedom to adapt to the personalization data, and so, effectively avoids overfitting at the cost of jeopardizing the personalization. This argument is strengthened with the results in Figure 6. In this figure, the individual accuracy improvements of 75 target subjects varies widely around the baseline zero line when $\alpha = 0$ and becomes more and more concentrated towards the zero baseline with increasing value of α. Apparently, a value around 0.4 is a reasonable choice for α.

Table 2 further provides a comparison of average performance obtained by personalization with different values of α with that before personalization. After personalization, the best performance is obtained with $\alpha = 0.4$, reaching an accuracy of 79.6% and improving over that of personalization without KL-divergence regularizaion and that of no-personalization by 2.2 and 4.5 percentage points absolute, respectively. Significant improvement on accuracy can also be seen from the confusion matrices in Figure 7 for most of the sleep stages. Furthermore, this accuracy level is on par with that of the model trained on the entire (known) cohort in Table 1 even though only one-night data of the subjects was used and the cohort was unknown.

Table 2: Average sleep staging performance before and after personalization. Personalization without KL-divergence regularization corresponds to $\alpha = 0$ and personalization with KL-divergence regularization corresponds to $\alpha > 0$. *All* finetuning was employed and personalization was run for 50 finetuning epochs.

		Overall metrics				
		Acc.	κ	MF1	Sens.	Spec.
Before personalization		75.1 ± 11.2	0.648 ± 0.140	67.2 ± 11.4	69.7 ± 11.4	93.1 ± 2.8
After personalization	$\alpha = 0$	77.4 ± 10.0	0.677 ± 0.131	71.4 ± 9.7	69.6 ± 10.8	93.6 ± 2.6
	$\alpha = 0.2$	79.0 ± 8.4	0.697 ± 0.114	72.5 ± 8.9	71.2 ± 10.2	94.0 ± 2.3
	$\alpha = 0.4$	$\mathbf{79.6 \pm 8.4}$	$\mathbf{0.706 \pm 0.113}$	$\mathbf{73.0 \pm 8.8}$	$\mathbf{71.8 \pm 10.1}$	$\mathbf{94.2 \pm 2.2}$
	$\alpha = 0.6$	78.8 ± 10.0	0.697 ± 0.128	72.0 ± 10.0	71.6 ± 10.9	94.0 ± 2.5
	$\alpha = 0.8$	77.0 ± 10.9	0.672 ± 0.138	69.2 ± 12.0	70.2 ± 11.8	93.5 ± 2.7

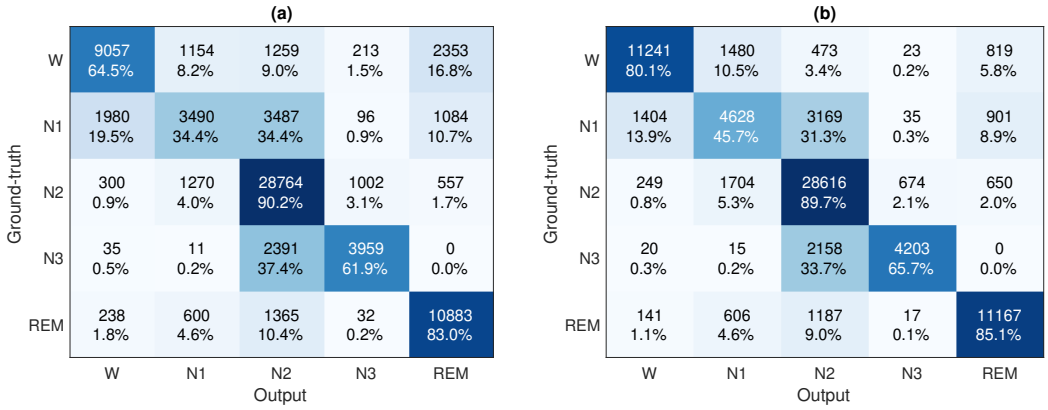

Figure 7: Confusion matrices obtained by SeqSleepNet before and after personalization. (a) Before personalization, (b) After personalization.

5.3. Influence of finetuning strategies.

It was shown in [7, 12] that, in sleep transfer learning, it is important to finetune feature-learning parts of a pretrained network to overcome the channel mismatch between a source domain and a target domain. This rule of thumb also applies to personalization as shown in Figure 5(b). Although finetuning the Softmax component alone brings up the performance, the improvement is significantly lower than the ones obtained by other finetuning strategies in which the feature-learning components of the pretrained SeqSleepNet (i.e. EPB or SBP or both) and the Softmax component are collectively adapted. For instance, the *All* finetuning strategy produces an accuracy improvement of 4.6 percentage points which is more than twice as much as the 1.9 percentage points obtained with the *Softmax* finetuning strategy after 50 finetuning epochs.

5.4. To personalize or not personalize?

In sleep transfer learning in general, when there is mismatch between the source domain (the MASS databased used for pretraining in our case) and the target domain (the personalization data in our case), it is vital to perform some form of finetuning. In case of personalization, besides possible discrepancies between the source-domain data and the personalization data [7], this data mismatch is further topped up with the target subject's peculiarities. On the contrary, when there is no data mismatch, finetuning could be averted as no significant improvement is expected while one increases the risk of overfitting. If there is a way to determine whether data distributions mismatch, one can decide to personalize the sleep staging model or not. Fortunately, we have access to the ground truth of a target subject's one-night data which can be utilized to assess the performance of the pretrained SI model. If the pretrained SI model performs well on this one-night data, the personalization data distribution is very likely matched to the source-domain data distribution. Reversely, poor performance of the pretrained SI model on this personalization data is an indicator of data mismatch.

In light of this observation, we applied a threshold β to the individual accuracy obtained on the first-night data to group 75 target subjects into two groups: Group A consisting of subjects with the accuracy before personalization below β and Group B consisting of subjects with accuracy before personalization equal or above β. Figure 8 shows the individual accuracies before personalization, the individual accuracies after personalization, and the individual accuracy improvements of the subjects in both groups with $\beta = 0.77$. As can be seen, most significant accuracy improvements correspond to the subjects in Group A while those improvements of the subjects in Group B are much more subtle. On average, personalization for Group A's subjects results in an improvement of 9.0 percentage points, ten times larger than that for Group B's subjects which is 0.9 percentage points.

6. Discussion

The personalization results in Figure 8 reveal uneven distribution of accuracy improvement across subjects. Those subjects on which the pretrained SI model performs poorly (i.e. severe data mismatch) benefit the most from personalization. However, only very modest improvements were seen for those subjects on which the pretrained SI model performs well, despite the fact that there is a similar channel mismatch: the C4-A1 EEG channel was used for pretraining the SI model and the Fpz-Cz EEG channel was used for personalization data. We speculate that personalization will be crucial for all target subjects when a completely different channel layout is used, for example in-ear EEG [9] or around-the-ear EEG [10, 11].

Setting a right value for the coefficient α was shown to play an important role in personalization's success. Although we have studied a common α for all target subjects and fixed its value during the personalization process, it makes more sense for α to be

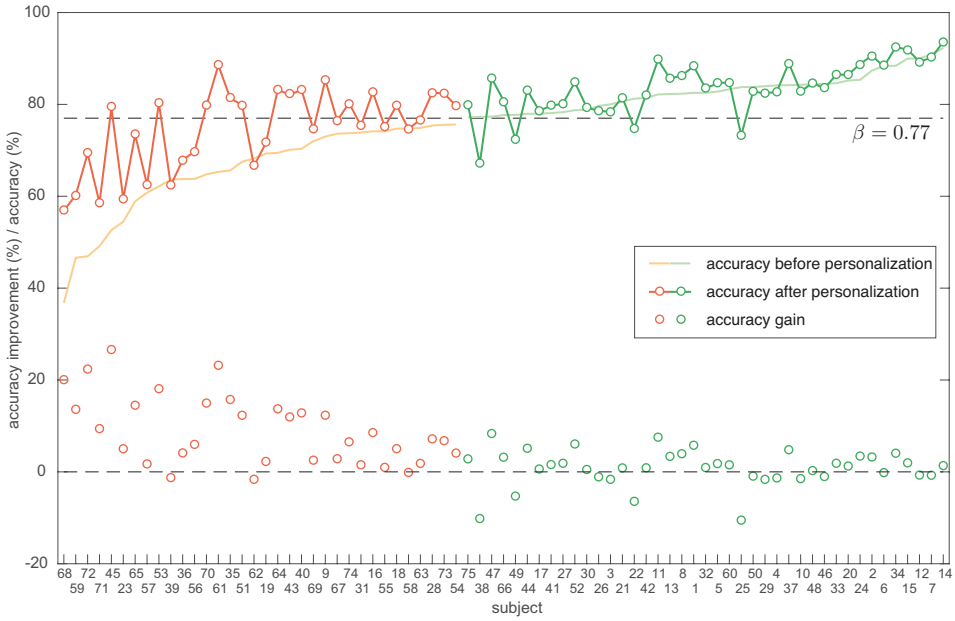

Figure 8: Individual accuracies before personalization, individual accuracies after personalization, and individual accuracy improvements of 75 target subjects (sorted by increasing accuracy before personalization). Group A (i.e. subjects with accuracy before personalization below β) is marked in orange and Group B (i.e. subjects with accuracy before personalization equal or greater than β) are marked in green. *All* finetuning was employed, α was fixed to 0.4, and personalization was run for 50 finetuning epochs.

adaptive. For example, for subjects with significant peculiarities (e.g. those subjects in Group A in Section 5.4), one should start with a large α to impose strong personalization inititally and attenuate it along the ongoing personalization process to gradually reduce this risk. The amount of personalization data should also be taken into account when setting a value for the KL-divergence regularization coefficient α. As a matter of fact, using single-night data for personalization is convenient. However, when more data is available, improvement on personalization performance can be expected. In intuition, α should be proportional to the amount of personalization data, i.e. we should use a small α for small personalization data (we trust the SI model more) and a large α for large personalization data (we trust the personalization data more).

7. Conclusions

We introduced the problem of sleep-staging personalization with single-night data and discussed its benefits and challenges in the context of personal sleep monitoring. We then attempted to tackle this problem using a transfer learning approach. The subject independent (SI) model (i.e. the pretrained SeqSleepNet) was used as the starting point and finetuned on the single-night data of a target subject to accomplish personalization. KL-divergence between the personalized model's output and the SI model's output is

proposed to regularize the network's loss function during personalization. The KL-divergence regularization anchors the personalized model, effectively preventing it from overfitting to the personalization data. Experimenting with 75 subjects of the Sleep-EDF Expanded database, we demonstrated that sleep personalization with a single-night data is possible. We showed that personalization implemented with KL-divergence regularization is robust against overfitting and achieves more favorable results compared to non-personalization and personalization without KL-divergence regularization.

In this pilot study, we demonstrated that automatic sleep staging with single-night data is possible and the obtained results are encouraging. However, while the number of subjects, at 75, is decently high, the population could still be considered quite homogeneous, which could impact the results shown. In addition, the fact that the database used in this study was labeled according to the old R&K guidelines [3] rather than the new and more robust AASM ones [2] may introduce some biases to the results. A larger database with diverging and richer characteristics (e.g. demographics, sleep diseases, and electrode placements etc.) is desirable for furture work. Such a database should be labeled (or re-relabeld by an independent sleep technician) following the AASM guidelines [2].

Acknowledgment

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

References

[1] J. M. Siegel, "Clues to the functions of mammalian sleep," *Nature*, vol. 437, no. 27, pp. 1264–1271, 2005.

[2] C. Iber, S. Ancoli-Israel, A. L. Chesson, and S. F. Quan, "The AASM manual for the scoring of sleep and associated events: Rules, terminology and technical specifications," *American Academy of Sleep Medicine*, 2007.

[3] J. A. Hobson, "A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects," *Electroencephalography and Clinical Neurophysiology*, vol. 26, no. 6, pp. 644, 1969.

[4] "National sleep research resource: free research data and tools," http://sleepdata.org/, assessed in 1 Jan 2020.

[5] C. O'Reilly, N. Gosselin, J. Carrier, and T. Nielsen, "Montreal archive of sleep studies: An open-access resource for instrument benchmarking & exploratory research," *Journal of Sleep Research*, pp. 628–635, 2014.

[6] J. B. Stephansen *et al.*, "Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy," *Nature Communications*, vol. 9, no. 1, pp. 5229, 2018.

[7] H. Phan, O. Y. Chén, P. Koch, Z. Lu, I. McLoughlin, A. Mertins, and M. De Vos, "Towards more accurate automatic sleep staging via deep transfer learning," *arXiv preprint arXiv:1907.13177*, 2019.

[8] A. Supratak, H. Dong, C. Wu, and Y. Guo, "DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.

[9] K. B. Mikkelsen, Y. R. Tabar, S. L. Kappel, C. B. Christensen, H. O. Toft, M. C. Hemmsen, M. L. Rank, M. Otto, and P. Kidmose, "Accurate whole-night sleep monitoring with dry-contact ear-EEG," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[10] K. B. Mikkelsen, J. K. Ebajemito, M. A. Bonmati-Carrion, N. Santhi, V. L. Revell, G. Atzori, C. della Monica, S. Debener, D.-J. Dijk, A. Sterr, and M. de Vos, "Machine-learning-derived sleep–wake staging from around-the-ear electroencephalogram outperforms manual scoring and actigraphy," *Journal of Sleep Research*, vol. 28, no. 2, pp. e12786, 2019.

[11] A. Sterr, J. K. Ebajemito, K. B. Mikkelsen, M. A. Bonmati-Carrion, N. Santhi, C. della Monica, L. Grainger, G. Atzori, V. Revell, S. Debener, D.-J. Dijk, and M. De Vos, "Sleep eeg derived from behind-the-ear electrodes (ceegrid) compared to standard polysomnography: A proof of concept study," *Frontiers in Human Neuroscience*, vol. 12, no. 452, 2018.

[12] H. Phan, O. Y. Chén, P. Koch, A. Mertins, and M. De Vos, "Deep transfer learning for single-channel automatic sleep staging with channel mismatch," in *Proc. 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[13] A. Agarwal, R. Dowsley, N. D. McKinney, D. Wu, C.-T. Lin, M. De Cock, and A. CA Nascimento, "Protecting privacy of users in brain-computer interface applications," *IEEE Trans. on Neural Systems and Rahabilitation Engineering*, vol. 27, no. 8, pp. 1546–1555, 2019.

[14] I. Martinovic, D. Davies, M. Frank, D. Perito, and T. Ros D. Song, "On the feasibility of side-channel attacks with brain-computer interfaces," in *Proc. 21st USENIX Security Symposium*, 2012.

[15] T. Bonaci, R. Calo, and H. J. Chizeck, "App stores for the brain: Privacy & security in brain-computer interfaces," *IEEE Technology and Society Magazine*, vol. 34, no. 2, pp. 32–39, 2015.

[16] K. B. Mikkelsen, D. B. Villadsen, M. Otto, and P. Kidmose, "Automatic sleep staging using ear-EEG," *BioMedical Engineering OnLine*, vol. 16, no. 111, 2017.

[17] K. Mikkelsen and M. De Vos, "Personalizing deep learning models for automatic sleep staging," *arXiv Preprint arXiv:1801.02645*, 2018.

[18] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging," *IEEE Transactions on Neural Systems and Rehabilitation Engineering (TNSRE)*, vol. 27, no. 3, pp. 400–410, 2019.

[19] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberye, "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG," *IEEE Trans. on Biomedical Engineering*, vol. 47, no. 9, pp. 1185–1194, 2000.

[20] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. e215–e220, 2000.

[21] S. A. Imtiaz and E. Rodriguez-Villegas, "An open-source toolbox for standardized use of PhysioNet Sleep EDF Expanded Database," in *Proc. EMBC*, 2015, pp. 6014–6017.

[22] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou, "Automatic sleep stage scoring with single-channel EEG using convolutional neural networks," *arXiv:1610.01683*, 2016.

[23] O. Tsinalis, P. M. Matthews, and Y. Guo, "Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders," *Annals of Biomedical Engineering*, vol. 44, no. 5, pp. 1587–1597, 2016.

[24] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Joint classification and prediction CNN framework for automatic sleep stage classification," *IEEE Trans. Biomedical Engineering (TBME)*, vol. 66, no. 5, pp. 1285–1296, 2019.

[25] S. A. Imtiaz and E. Rodriguez-Villegas, "Recommendations for performance assessment of automatic sleep staging algorithms," in *Proc. EMBC*, 2014, pp. 5044–5047.

[26] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "DNN filter bank improves 1-max pooling CNN for single-channel EEG automatic sleep stage classification," in *Proc. EMBC*,

2018, pp. 453–456.

[27] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Automatic sleep stage classification using single-channel EEG: learning sequential features with attention-based recurrent neural networks," in *Proc. EMBC*, 2018, pp. 1452–1455.

[28] F. Andreotti, H. Phan, N. Cooray, C. Lo, M. T. M. Hu, and M. De Vos, "Multichannel sleep stage classification and transfer learning using convolutional neural networks," in *Proc. EMBC*, 2018, pp. 171–174.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computing*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville, "Recurrent batch normalization," *arXiv Preprint arXiv:1603.09025*, 2016.

[31] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, 2015, pp. 1412–1421.

[32] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. ICASSP*, 2013, pp. 7893–7897.

[33] S. Mousavi, F. Afghah, and R. Acharya, "SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach," *PLoS One*, vol. 14, no. 5, pp. e0216456, 2019.

[34] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv:1603.04467*, 2016.

[35] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–13.

[36] Y. Yang, , and X. Liu, "A re-examination of text categorization methods," in *Proc. SIGIR*, 1999, vol. 99, pp. 42–49.

[37] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia Medica*, 2012.